Regularized Online Mixture of Gaussians for Background Subtraction

Wang, H., & Miller, P. (2011). Regularized Online Mixture of Gaussians for Background Subtraction. In Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on. (pp. 249-254). Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/AVSS.2011.6027331

Published in:
Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on

Regularized Online Mixture of Gaussians for Background Subtraction

Hongbin Wang and Paul Miller
The Centre for Secure Information Technologies (CSIT)
Queen's University of Belfast, Belfast, BT3 9DT, UK
{h.wang, p.miller}@ecit.qub.ac.uk

Abstract

Mixture of Gaussians (MoG) modelling [13] is a popular approach to background subtraction in video sequences. Although the algorithm shows good empirical performance, it lacks theoretical justification. In this paper, we give a justification for it from an online stochastic expectation maximization (EM) viewpoint and extend it to a general framework of regularized online classification EM for MoG with guaranteed convergence. By choosing a special regularization function, l_1 norm, we derived a new set of updating equations for l_1 regularized online MoG. It is shown empirically that l_1 regularized online MoG converge faster than the original online MoG .

1. Introduction

Background subtraction is an important step in video analytics for surveillance. The Mixture of Gaussians (MoG) model [13] is one of the most popular algorithms for this and shows good empirical performance. Most of the extensions and improvements [2] of the original method focus on various practical issues such as dynamic backgrounds, shadows and illumination changes. Up to now, only a few papers have focused on the theoretical aspects of the MoG method. In a pioneering work [5], the authors derive a set of updating equations for MoG using an incremental version of EM [9]. The equations try to remember all of the past information and have no learning rate parameter, making them different to the equations in [13]. In other work [11], the authors start from a batch version of EM for MoG and moves to online updating equations using an online averaging viewpoint, for which is not easy to get theoretical justification.

In this paper we derive the original set of equations using online EM and stochastic approximation [1]. Furthermore, we propose a general framework of regularized online classification EM for MoG, which includes the original online MoG [13] as a special case without a regularization term. The main purpose of this paper is to put the online MoG algorithm on a solid footing, from a theoretical viewpoint. In the following sections, we firstly justify the original MoG algorithm, then extend it to a general framework with a regularized term, and finally investigate a special case, namely, the l_1 regularization function.

1.1. Background subtraction with mixture of Gaussians (MoG)

The MoG background subtraction algorithm has two components; the online MoG algorithm for pixel density estimation, and binary classification to decide whether each pixel is background or foreground. The details are as follows. Given a video sequence, MoG maintains a parametric density function P_t for each pixel at time t. The value of a pixel at time t is denoted by x_t. Here we assume x is gray-scale intensity and can be directly extended to multi-dimensional color or other feature spaces if we assumes a diagonal covariance matrix. The pixel distribution $P_t(x)$ is modelled as a mixture of K Gaussians:

$$P_t(x|\theta_t) = \sum_{i=1}^{K} w_{i,t} G(x|\mu_{i,t}, \sigma_{i,t}) \tag{1}$$

where $\theta_t = \{w_{i,t}, \mu_{i,t}, \sigma_{i,t}\}_{i=1}^{K}$, $G(x, \mu_{i,t}, \sigma_{i,t})$ is the i-th Gaussian component with mean $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$. $w_{i,t}$ is the weight of the i-th Gaussian component. Typically, K ranges from three to five. For each new pixel value x_t, a match is found if $|x_t - \mu_{i,t-1}| \leq f\sigma_{i,t-1}$, for any $i = 1, 2, \cdots, K$. In practice f is usually set to 2.5.

If a match is found, the parameters of MoG are updated as follows:

$$w_{i,t} = (1 - \alpha) w_{i,t-1} + \alpha \tag{2}$$

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho x_t \tag{3}$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(x_t - \mu_{i,t})^2 \tag{4}$$

where α is the learning rate for the weight and ρ is the learning rate for the distribution.

After $P_t(x|\theta_t)$ is obtained, the Gaussians are ranked according to their associated term w/σ. The background is then modelled by the first B largest Gaussians chosen as follows:

$$B = \arg\min_b (\sum_{i=1}^{b} w_i > T) \tag{5}$$

where T denotes the portion of the data that we assume belongs to the background.

So the background model $P_t(x|BG)$ is given by

$$P_t(x|BG) = \sum_{i=1}^{B} w_{i,t} G(x, |\mu_{i,t}, \sigma_{i,t}) \tag{6}$$

To summarise: For each new video frame, the MoG algorithm will perform the following three steps :

1. Perform binary classification based on background model $P_t(x|BG)$ given by equation 6.
2. Update $P_t(x|\theta_t)$ using equations 2, 3 and 4.
3. Update $P_t(x|BG)$ using equation 5.

It is clear that step 2 is the key to the algorithm, as the other two steps depend upon it. In this paper, we mainly focus on the online MoG algorithm. To begin, we first answer the question: Where do these equations come from?

1.2. Online EM

To answer the above question, let us step back and recall the EM algorithm [4] for MoG, then extend to the online version. We will provide the link to online MoG algorithm in the next section.

Given a MoG model $p(x|\theta)$, the maximum likelihood estimation of its parameter θ can be obtained by

$$\theta = \arg\max_\theta [\log p(X|\theta)]$$

Where the set of all observed data is denoted by X. Let us introduce a K-dimensional latent variable z to indicate the assignment of x to one of K components. If data point x belongs to component k, $z_k = 1$. Otherwise $z_k = 0$. The set of all latent variable is denoted by Z, so $\log p(X|\theta) = \log\{\sum_Z p(X, Z|\theta)\}$.

The well known EM algorithm runs iteratively and has two steps; an E step followed by an M step. In the E step, it uses the old parameter θ^{old} to estimate the conditional distribution of the latent variable $p(Z|X, \theta^{old})$. In the M step, the parameter can be updated by maximizing the function

$$\theta^{new} = \arg\max_\theta Q(\theta, \theta^{old})$$

Where $Q(\theta, \theta^{old})$ is the expectation of the complete data log likelihood

$$Q(\theta, \theta^{old}) = \sum_Z p(Z|X, \theta^{old}) \log[p(X, Z|\theta)]$$

In online mode, only one data point is observed at a time. So the data set X only contains one point x_t at a time, hense the log likelihood function is changed to $\log p(x_t|\theta)$.

In practice, the M step may be intractable. So instead of trying to maximize $Q(\theta, \theta_{t-1})$, it tries to increase its value. This is also the principle of generalized EM [9]. There exist two main variants of online EM algorithm [8]. One is incremental EM [9], which remembers all of the past sufficient statistics to update the parameters. The other, proposed in [3] , forgets older sufficient statistics as time progresses. Incremental EM is inappropriate for background modeling application as it has linearly growing memory usage, and its stored older sufficient statistics may not help updating the future parameters [8]. So we turn our focus to the learning rate approach, which increases the value of $Q(\theta, \theta_{t-1})$ by a stochastic approximation step [1]

$$\theta_t = \theta_{t-1} + \gamma_t \nabla Q(\theta, \theta_{t-1}) \tag{7}$$

where $\gamma_t > 0$ is learning rate and $Q(\theta, \theta_{t-1})$ is the expectation of complete-data log likelihood

$$Q(\theta, \theta_{t-1}) = \sum_Z p(Z|x_t, \theta_{t-1}) \log[p(x_t, Z|\theta)]$$

In [3] a proof was given that, under the conditions $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$, the algorithm will converge to a local minimum. We call this *online stochastic EM* and it is shown in Algorithm 1.

Algorithm 1 - *Online stochastic EM*:
Given an observed sequence $x_1, \ldots, x_t$ and denoting the parameter set as θ.
1. Initialize parameter set to θ_0
At each time stamp t, perform the following steps with the new observation x_t:
2. E step: to calculate the conditional distribution $p(Z|X, \theta_{t-1})$.
3. M step: to update parameter set by equation (7).
4. if not convergent, set $t \leftarrow t + 1$ and return to step 2.

---

2. The justification of MoG background subtraction algorithm

In [13] the explanation given for their approach was "... to use the following on-line K-means approximation to update the mixture model". However, they omitted to give a

detailed derivation. In this section, we give an explanation for online MoG based on the online stochastic EM algorithm above.

First of all, comparing K-means to the EM algorithm for the mixture modelling, the former does a hard assignment of a data point to a centre, whereas EM does a soft assignment based on the conditional probability $p(Z|X, \theta_{t-1})$. However, "on-line K-means approximation" means the EM algorithm does a hard assignment for the mixture model. This can be done by introducing a classification step (C step) between the E and M steps to uniquely classify the data point x_t to one of data centers k [15], where $k = \arg\max_Z [\log p(Z|x_t, \theta_{t-1})]$.

Next we rewrite the E step in algorithm 1 using the conditional distribution of MoG as

$$p(Z = i|X, \theta_{t-1}) = \frac{w_{i,t-1} G(X|\mu_{k,t-1}, \sigma_{k,t-1})}{\sum_i w_{i,t-1} G(X|\mu_{k,t-1}, \sigma_{k,t-1})}$$

After classifying x_t into its optimal Gaussian component, the log likelihood of a matched class k in the M step is

$$Q(\theta, \theta_{t-1}) = \log(w_{k,t-1} G(x_t|\mu_{k,t-1}, \sigma_{k,t-1}))$$

We then calculate the equation in M step to get

$$w_{k,t} - w_{k,t-1} = \gamma_{w,t}(1 - w_{k,t-1}) \tag{8}$$

$$\mu_{k,t} - \mu_{k,t-1} = \gamma_{\mu,t}(x_t - \mu_{k,t-1}) \tag{9}$$

$$\sigma^2_{k,t} - \sigma^2_{k,t-1} = \gamma_{\sigma,t}[\sigma^2_{k,t-1} - (x_t - \mu_{k,t})^2] \tag{10}$$

The learning rate γ_t is the step size in the gradient direction [1]. If we let $\gamma_{w,t} = \alpha$ and $\gamma_{\mu,t} = \gamma_{\sigma,t} = \rho$, we get the same updating equations as those in (2), (3) and (4).

In summary, the online MoG algorithm [13] can be viewed as an online classification EM algorithm for Mixture of Gaussians.

3. Regularized online classification EM

The online stochastic EM algorithm takes a stochastic approximation step to update the parameter set θ. This noisy approximation limits the convergence speed [1] and may cause the algorithm to be unstable [15]. One of solutions is to introduce a regularized term on θ , thereby constraining its behaviour. The application of this idea to online learning starts in paper [6]. Here we add a regularized term into the maximization equation of online classification EM.

$$\theta_t = \theta_{t-1} + \gamma_t \nabla Q(\theta, \theta_{t-1}) - \beta_t \nabla R(\theta) \tag{11}$$

Where β_t is a regularization parameter, $R(\theta)$ is the regularization term and $Q(\theta, \theta_{t-1})$ is the log likelihood of the matched class. From a cost function minimization viewpoint, let's denote a cost function $C(\theta, \theta_{t-1}) = -Q(\theta, \theta_{t-1}) + R(\theta)$, where the first term is the negative log-likelihood function and the second term is the regularization function. So the M step becomes

$$\theta_t = \arg\min_\theta C(\theta, \theta_{t-1})$$

And if using a stochastic gradient step to minimize $C(\theta, \theta_{t-1})$, we will get equation (11) .

Algorithm 2 - *Regularized online classification EM*:
Given an observed sequence $x_1, \ldots, x_t$ and denoted the set of parameter as θ.
1. Initialize the parameter to θ_0
At each time stamp t, perform the following steps with the new observation x_t:
2. E step: to calculate the conditional distribution $p(Z|X, \theta_{t-1})$.
3. C step: to assign the data point x_t to class k, where $k = \arg\max_Z \log p(Z|x_t, \theta_{t-1})$
4. M step: to update parameter set by equation (11)
5. if not convergent, set $t \leftarrow t + 1$ and return to step 2.

The proposed regularized online classification EM algorithm, listed in Algorithm 2, is different with previous work. Compared to other online EM algorithms [15, 3] , we generalize these by introducing a regularized term into the maximization step, making the algorithm more flexible. Compared to online convex optimization algorithms [6, 7], our algorithm targets the non-convex optimization problem. By adding a classification step into EM, we simplify a non-convex optimization problem into several smaller convex optimization problems. For example, in the MoG case, the cost function of MoG is non-convex. However, by applying (regularized) online classification EM, after classifying the latent variable Z, the M step only considers a single Gaussian problem, which has a convex cost function.

Examples of the regularization term $R(\theta)$ include:

- l_1 regularization: $R(\theta) = ||\theta||_1$, where $||\bullet||_1$ is 1-norm operator
- l_2 regularization: $R(\theta) = ||\theta||_2^2$, where $||\bullet||_2$ is 2-norm operator
- Kullback-Leibler(KL) divergence regularization: The KL-divergence $D_{KL}(P||Q)$ normally is used to mea-

sure the difference between two probability distributions P and Q. It is defined as

$$D_{KL}(P||Q) = \int_{x \in R^d} P(x|\theta) \log \frac{P(x|\theta)}{Q(x|\theta)} dx$$

4. l_1 regularized online MoG and online heavy-ball method

In this section, we choose l_1 regularization and investigate it further. The l_1 regularization is widely used to produce sparse solutions and examples of its application in online learning also exist [7, 12]. In high dimensional learning problems [7, 12], l_1 regularization produces a sparse result by forcing most of the parameters to zero. In low dimensional case, and under some conditions, we propose that, instead of producing sparsity, l_1 regularization accelerates the convergence.

In video background subtraction, the image pixel value $x \geq 0$, hence, then the associated parameter set $\theta_t = \{w_{i,t}, \mu_{i,t}, \sigma_{i,t}\}_{i=1}^K \geq 0$. So the added l_1 regularization term can be simplified to $R(\theta) = \theta$. This means that adding the l_1 regularization term in background subtraction is equivalent to adding a linear term. Another difference with the traditional usage of l_1 regularization, is that we do not require the regularization parameter β_t non-negative. This is because we are only concerned with the convexity of the cost function and do not try to force sparsity in the solution. By substituting into the M step in Algorithm 2 we obtain

$$\theta_t = \theta_{t-1} + \gamma_t \nabla Q(\theta, \theta_{t-1}) - \beta_t$$

If we choose $\beta_t = \beta(\theta_{t-2} - \theta_{t-1})$, then

$$\theta_t = \theta_{t-1} + \gamma_t \nabla Q(\theta, \theta_{t-1}) + \beta(\theta_{t-1} - \theta_{t-2}) \qquad (12)$$

The l_1 regularization term finally becomes $\beta(\theta_{t-1} - \theta_{t-2})$, where β is a time dependent parameter which we choose as a diminishing sequence $\left\{\frac{1}{t}, t = 1, \cdots n\right\}$, and $\theta_{t-1} - \theta_{t-2}$ is the updating amount from the previous step. This two step method, which was proposed for batch optimization [10], is called the heavy-ball method or gradient method with momentum.

For batch optimization, it can be shown that the heavy-ball method has better convergence speed than the gradient descent method [10, 14]. Let's denote the local optimal point θ^*. The Lyapunov function $||\theta_k - \theta^*||$ of the heavy-ball method has an approximately linear convergence rate $\left(1 - \frac{2}{\sqrt{\kappa}}\right)$, while the gradient descent method is with $(1 - \frac{2}{\kappa})$, where κ is a problem dependent constant. It was shown [14], that if we want $||\theta_k - \theta^*||$ reduced by a factor ϵ, then the heavy-ball method needs at least $\frac{\sqrt{\kappa}}{2} \log \epsilon$ steps, while the gradient descent method needs at least $\frac{\kappa}{2} \log \epsilon$ steps. This is a $\sqrt{\kappa}$ difference, which means, if $\kappa = 100$, the heavy-ball method needs ten times less steps than the gradient descent method.

For the online optimization case here, by adding a linear term into the cost function, we ensure convexity holds. So the proof of convergence in [3] is still valid.

When we apply it to Mixture of Gaussian case, the updating equations are as follows:

$$\begin{aligned} w_{k,t} - w_{k,t-1} &= \gamma_{w,t}(1 - w_{k,t-1}) \\ &\quad +\beta(w_{k,t-1} - w_{k,t-2}) \end{aligned} \qquad (13)$$

$$\begin{aligned} \mu_{k,t} - \mu_{k,t-1} &= \gamma_{\mu,t}(x_t - \mu_{k,t-1}) \\ &\quad +\beta(\mu_{k,t-1} - \mu_{k,t-2}) \end{aligned} \qquad (14)$$

$$\begin{aligned} \sigma^2_{k,t} - \sigma^2_{k,t-1} &= \gamma_{\sigma,t}[\sigma^2_{k,t-1} - (x_t - \mu_{k,t})^2] \\ &\quad +\beta(\sigma^2_{k,t-1} - \sigma^2_{k,t-2}) \end{aligned} \qquad (15)$$

5. Experiments

For the first experiment simulated data was used to see how the proposed l_1 regularized algorithm behaves compared to the original MoG. The simulated data is a three component MoG with parameters: means $\{60, 120, 180\}$, variances $\{81, 81, 81\}$ and weights $\{0.5, 0.3, 0.2\}$. The initial parameters are $\{w_{k,0} = 0.1, \mu_{k,0} = 0, \sigma^2_{k,0} = 400, k = 1, 2, 3\}$ and the learning rate $\gamma = 0.01$. The regularization parameter β in l_1 regularized online MoG is a time dependent sequence $\{\frac{1}{t}, t = 1, \cdots n\}$. The experiment is repeated 500 times with each run generating 200 data samples. The convergence results of the parameters are averaged over the 500 runs.

The convergence results for all three components are quite similar, so here we only show the result of the component with $\{w = 0.3, \mu = 120, \sigma^2 = 81\}$. The convergence result of these parameters are shown in Figure 1. Comparing the original online MoG with the proposed l_1 regularized algorithm, the mean converges faster, Fig. 1(c), whilst the variance converges slower, Fig. 1(d). The weight has almost the same convergence curve for both, Fig. 1(b). In the mean parameter updating process, the contribution from the previous step in the l_1 regularized algorithm is significant, as it pushes the mean into a saturated level at around 110 using approximately twenty samples, whilst the original online MoG needs approximately eighty samples to reach the same level. The story of the variance for the l_1 regularized algorithm is a little different. Because the initial stage of the

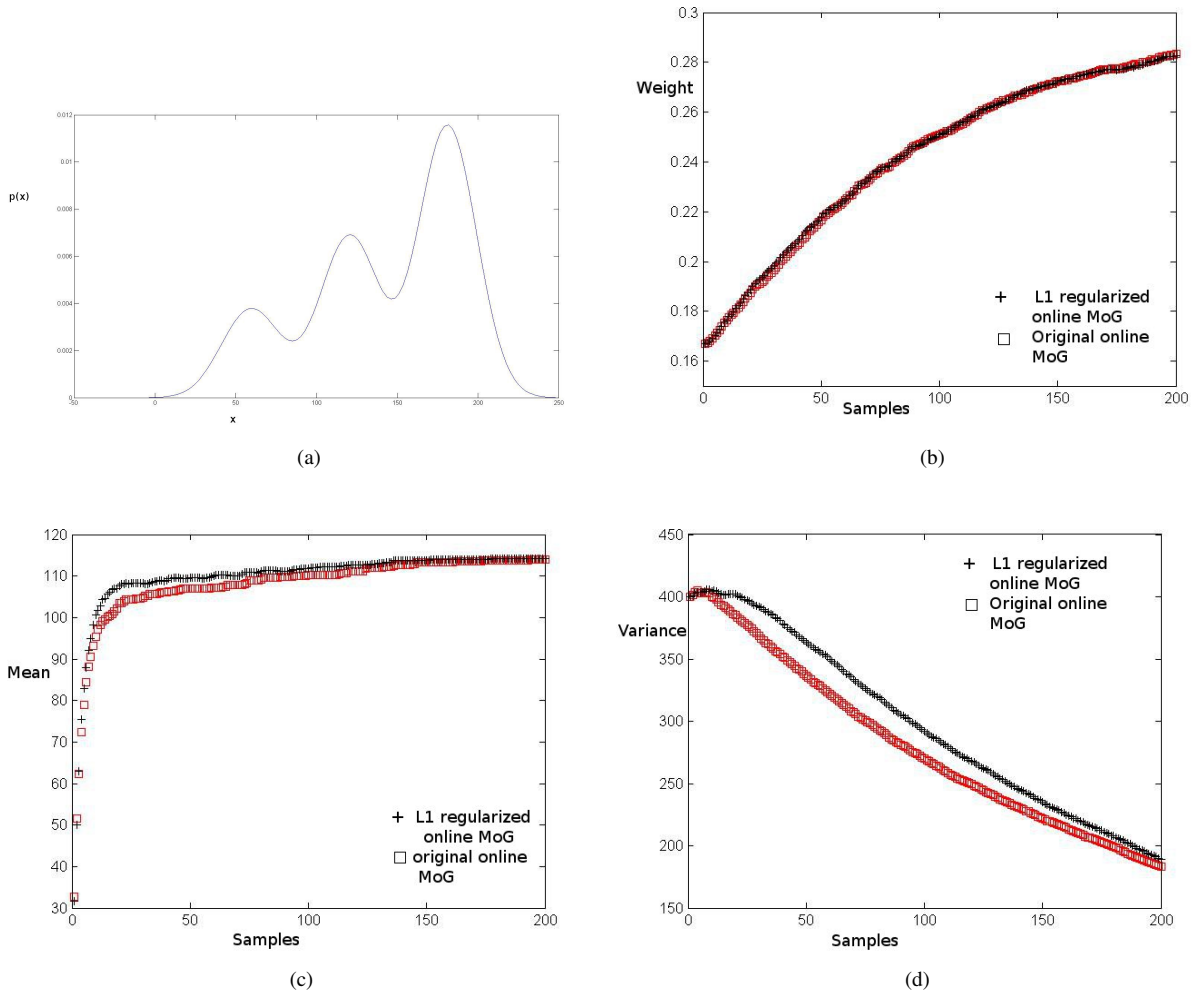

Figure 1: Simulated data of a three component MoG, (a). Convergence plots for the weight, (b), mean, (c), and variance, (d), parameters.

variance updating tends to increase the value, instead of decreasing towards the ideal value, the contribution from the previous step is negative. This causes the variance to update slower than the original algorithm.

In addition to simulations, we also carried out some background subtraction experiments on real data obtained from the PETS 2001 camera two video data set. The sequence consists of 2688 frames of size 768 x 576 pixels. Figure 2 (a) shows a representative frame and the background subtraction output, Fig. 2(c) and (d). The sequence has been manually annotated in that for each frame a bounding box has been manually placed around the moving objects. We take the bounding box to give a reasonable approximation to the ground truth, Fig. 2(b). We calculate the ROC curves after doing binary classification using the learned background model for both algorithms with equation (7). The true positive rate and false positive rate are averaged over the 2688 frames. The ROC result in Figure 3 shows that the proposed l_1 regularized online MoG has better performance. This is supported by the simulation result above. Faster mean convergence leads to better region selection of the moving object. Slower variance convergence makes the existing Gaussian components more stable, resulting in new components being introduced infrequently.

6. Conclusion

We propose a general framework of regularized online classification EM for MoG, which includes the original online MoG [13] as a special case without regularization. We also investigate a specific case that employs l_1 regularization. Compared to the original online MoG, it is shown em-

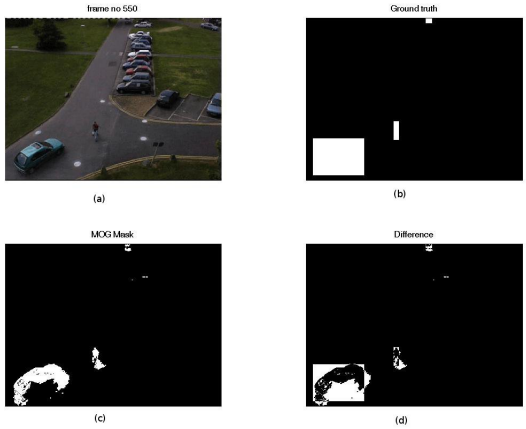

Figure 2: A example frame of PET2001 sequence (a), corresponding ground truth (b) and its background subtraction output (c) and (d).

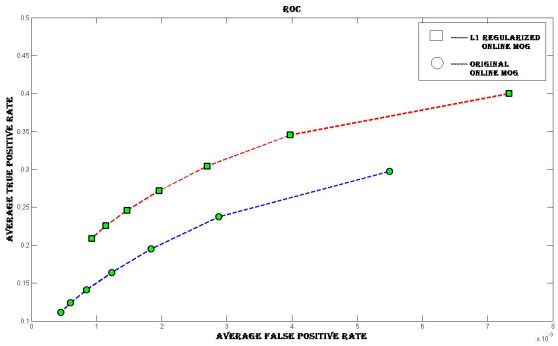

Figure 3: Comparison of ROC curves of l_1 regularized online MoG and original online MoG

pirically that the l_1 regularized online MoG has faster mean convergence and slower variance convergence, which leads to a better empirical background subtraction result with the PETS 2001 benchmark data set. In future we'd like to investigate online classification EM with other regularization functions such as l_2 norm and KL divergence.

Acknowledgement

This research work is sponsored by the EPSRC projects EP/E028640/1 and EP/G034303/1.

References

[1] Léon Bottou: Online Algorithms and Stochastic Approximations, Online Learning and Neural Networks, Edited by David Saad, Cambridge University Press, Cambridge, UK, 1998.

[2] T. Bouwmans, F. El Baf and B. Vachon. Background Modeling using Mixture of Gaussians for Foreground Detection – A survey. Recent Patents on Computer Science, Vol. 1, No 3, pp.219-237, Nov. 2008.

[3] Cappé, O. and Moulines, E. On-line expectation-maximization algorithm for latent data models. J. Roy. Statist. Soc. B, 71(3):593–613. 2009.

[4] Dempster, A. P., N. M. Laird, and D. B. Rubin . Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B 39(1), 1–38. 1977.

[5] Friedman N, Russell S. Image Segmentation in Video Sequences: A Probabilistic Approach. Proceedings Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI 1997), 1997.

[6] Jyrki Kivinen and Manfred K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. Information and Computation, 132(1):1–64, January 1997.

[7] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. Journal of Machine Learning Research, 10:777–801, 2009.

[8] Liang, P. and Klein, D. Online EM for unsupervised models. In Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). 2009.

[9] R.M. Neal and G.E. Hinton, A view of the EM algorithm that justifies incremental, sparse and other variants. in Learning in Graphical Models edt. M.I. Jordan, pp. 355-368, Kluwer academic press, Norwell, 1998.

[10] Polyak, B. T. Introduction to Optimization, Optimization Software Inc. 1987.

[11] P. W. Power and J. A. Schoones. Understanding background mixture models for foreground segmentation. in Imaging and Vision Computing, Nov. 2002.

[12] S. Shalev-Shwartz and A. Tewari. Stochastic methods for l_1 regularized loss minimization. ICML09, 2009.

[13] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. in IEEE Trans. on PAMI, 22, pp. 747-757, Aug 2000.

[14] S. J. Wright. Optimization Algorithms in Machine Learning. NIPS Tutorial, December, 2010.

[15] J.-F. Yao. On recursive estimation of incomplete data models. Statistics, 34(1):27-51, 2000.